What is claimed is:

1. A computer-implemented method of initializing a simulation comprising:

accessing an attribute on a simulation model; and

responsive to the attribute, conditionally treating an unknown state of an input node on the simulation model as a known state.

2. The computer-implemented method of claim 1 wherein conditionally treating an unknown state comprises:

when the attribute is set, evaluating an output node of the simulation model using the known state for the input node rather than the unknown state.

3. The computer-implemented method of claim 2 wherein evaluating an output node comprises detecting an X on the input node, and evaluating the output node as if a 0 was on the input node.

4. The computer-implemented method of claim 3 wherein the simulation model is a model of an inverter, and evaluating an output node comprises detecting an X on the input node, and evaluating the output node to a 1.

5. The computer-implemented method of claim 2 wherein evaluating an output node comprises detecting an X on the input node, and evaluating the output node as if a 1 was on the input node.

6. The computer-implemented method of claim 5 wherein the simulation model is a switch-level model of a transistor, and evaluating an output node comprises detecting an X on a gate node of the switch-level model of the transistor, and logically closing the switch-level model of the transistor.

7. A computer-implemented method of evaluating an output node of a device model comprising:

determining whether an input node of the device model has an unknown value assigned thereto; and

responsive to an attribute on the device model, if the input node has an unknown value, conditionally evaluating the output node as if the input node had a known value.

8. The computer-implemented method of claim 7 wherein the attribute has at least two valid states comprising a first state signifying that the unkown value should be propagated to the output node, and a second state signifying that the unknown state should be treated as a known state, and wherein conditionally evaluating comprises:

when the attribute is in the first state and the input node has an unknown value, propagating the unknown value to the output node.

9. The computer-implemented method of claim 8 further comprising:

when the attribute is in the second state and the input node has an unknown value, evaluating the output node as if the input node had a known value thereon.

10. The computer-implemented method of claim 7 wherein the method is performed within a three value simulator that represents node values by a two bit word, and wherein evaluating the output node comprises:

inverting one bit of the two bit word representing the unknown value on the input node to create a temporary two bit word; and

evaluating the output node using the temporary two bit word as the input node value.

11. A computer-implemented method of simulating a self-resetting circuit, the method comprising:

detecting the presence of an X value on a node within the self-resetting circuit;

accessing an attribute on a device model that represents a device within the self-resetting circuit, wherein the node having the X value is an input node of the device; and

when the attribute is set, simulating the self-resetting circuit as if the node had a non-X value.

12. The computer-implemented method of claim 11 wherein simulating the self-resetting circuit as if the node had a non-X value comprises simulating the self-resetting circuit as if the node had a value of zero.

13. The computer-implemented method of claim 12 wherein the device comprises an inverter, and simulating comprises evaluating an output node of the inverter to a value of one.

14. The computer-implemented method of claim 11 wherein the X value is represented by a two bit word, and simulating the self-resetting circuit as if the node had a non-X value comprises inverting at least one bit of the two bit word.

15. The computer-implemented method of claim 14 wherein the device model represents a first inverter in an inverter chain, and simulating the self-resetting circuit further comprises exchanging the values of the bits in the two bit word to create a two bit word that represents the value of an output node of the device model.

16. An article having a computer readable medium, the computer readable medium having instructions stored thereon for performing a method of initializing a device model in a simulation, the method comprising:

accessing an attribute of the device model to ascertain a state of the attribute; and

responsive to the state of the attribute, conditionally treating an X on an input node of the device model as a value other than an X.

17. The article of claim 16 wherein the attribute is associated with the input node, the method further comprising:

accessing a second attribute of the device model, the second attribute being associated with a second input node; and

responsive to a state of the second attribute, conditionally treating an X on the second input node as a value other than an X.

18. The article of claim 16 wherein conditionally treating comprises:

conditionally treating an X on any input node of the device model as a value other than an X.

19. The article of claim 16 wherein the device model is a model of an inverter, the simulation is a switch-level simulation, and conditionally treating comprises:

when the attribute is set and an X is present on an input node to the model of the inverter, evaluating an output node of the model of the inverter to a 1.

20. The article of claim 16 wherein the device model is a model of an inverter, the simulation is a switch-level simulation, and conditionally treating comprises:

when the attribute is set and an X is present on an input node to the model of the inverter, evaluating an output node of the model of the inverter to a 0.

21. An article having a computer readable medium, the computer readable medium comprising a data structure describing a device model for use in a simulator, the data structure comprising an attribute to signify whether an X on an input node of the device model should be treated as a value other than X during a simulation.

22. The article of claim 21 wherein the attribute is associated with the device model such that the attribute is a single attribute configured to affect a simulator's behavior relative to all input nodes of the device model.

23. The article of claim 21 wherein the attribute is associated with the input node of the device model such that the attribute is configured to affect a simulator's behavior relative to only the input node to which the attribute is associated.

24. The article of claim 23 wherein the data structure further comprises a second attribute associated with a second input node of the device model, the second attribute being configured to signify whether an X on the second input node of the device model is to be treated a value other than X during simulation.

25. The article of claim 21 wherein the device model represents an inverter, and the device model is configured to be used in a switch-level simulator.